

Министерство науки и высшего образования Российской Федерации
Пензенский государственный университет
Кафедра «Системы автоматизированного проектирования»

Курсовая работа по дисциплине
«Информационное обеспечение систем автоматизированного
проектирования»
на тему «Проектирование БД и разработка приложения учёта
использования программных средств»
Направление подготовки 09.03.01 «Информационная и
вычислительная техника»
профиль «Системы автоматизированного проектирования»

Выполнил ст. группы 18ВВЗ
Канеев Э.К.

Приняли:
к.т.н. доцент Глотова Т.В.
к.т.н. Евсеева Ю.И.

Пенза 2020

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра _____

УТВЕРЖДАЮ

Зав. кафедрой _____

“ ” _____ г.

[Подпись]

ЗАДАНИЕ

на курсовое проектирование по курсу

По курсу Информационное обеспечение САПР

Студенту Каневу Э.К.

Группа 18ВВЗ

Тема проекта

Проектирование БД и разработка приложения
учёта использования программных средств

Исходные данные (технические требования) на проектирование

- информационное проектирование должно быть выполнено с помощью CASE-средств.
- на этапе логического проектирования обязательна проверка на соответствие нормальным формам отношений.
- разработанное приложение должно выполнять функции ввода, обновления, добавления, поиска данных по ключу, сервисные функции, копирования.
- использование операторов языка SQL.
- запросы к базе данных должны быть оптимизированы.
- во время ввода и измерения база данных должно проводиться проверки на целостность базы данных.
- организационная интерпретация пользователя должна отвечать требованиям стандарта среды Windows.
- количество записей тестовых примеров базы данных должно быть достаточно для полного тестирования приложения.
- используемая СУБД - исходные данные .4d

Объем работы по курсу

1. Расчетная часть

1. Проектирование иерархической модели БД (концептуальной схемы)
2. Проектирование логической схемы БД
3. Проектирование физической схемы БД
4. Разработка приложений
5. Разработка структуры приложения

2. Графическая часть

1. Схема "Сущность-связь" для описания иерархической модели БД
2. Схема структуры приложения

3. Экспериментальная часть

1. Отладка приложения

Срок выполнения проекта по разделам

1	Анализ ТЗ и 1.1	к	23 сентября
2	1.2	к	30 сентября
3	1.3	к	7 октября
4	1.4	к	7 октября
5	1.5	к	4 ноября
6	3.1	к	27 ноября
7	Оформление П.З.	к	9 декабря
8		к	

Дата выдачи задания "9" сентября 2020 г.

Дата защиты проекта "9" декабря 2020 г.

Руководитель

Задание получил "9" сентября 2020 г.

Студент

Ван

Содержание

Введение	5
1 Инфологическое проектирование.....	6
1.1 Инфологическая модель.....	6
1.2 Связи	8
2. Логическое проектирование	10
2.1 Нормализация	10
2.2 Соответствие нормальным формам.....	12
3. Физическое проектирование.....	14
3.1 Типы данных.....	14
3.2 Индексы	16
4 Разработка структуры приложения.	19
5 Разработка приложения	21
5.1 Разработка запросов	21
5.2 Разработка программных модулей	27
5.3 Интерфейс работы с приложением.....	29
Заключение	34
Список используемой литературы	35
Приложение А	36
Приложение Б.....	44

Введение

В современном обществе объем информации очень большой и часто требуется структурирование этой информации для применения в какой-либо области деятельности. Использование информационных систем решает данную проблему. Основой любой информационной системы является база данных.

Для разработки Web-приложения был использован язык PHP. PHP - язык программирования, специально разработанный для написания web-приложений (скриптов, сценариев), исполняющихся на Web-сервере.

Целью курсовой работы является разработка базы данных для учёта ПО.

Для достижения поставленной цели были определены следующие задачи:

- изучить СУБД MYSQL
- изучить скриптовый язык PHP
- разработать структуру базы данных «Учёт ПО»,
- создать базу данных в среде MYSQL.
- создать базу данных в среде phpMyAdmin.
- создать скрипты на PHP для работы с базой данных
- разработать Web-интерфейс для работы с базой данных

1 Инфологическое проектирование

1.1 Инфологическая модель

Предметная область – Программное обеспечение в ВУЗе.

База данных должна содержать сведения о следующих объектах: кафедра, аудитория, компьютеры, характеристики компьютера, требования ПО, ПО, договор, лицензия, категории ПО.

На основе анализа предметной области и возможных запросов пользователей (студенты, преподаватели) была разработана инфологическая модель БД(рис. 1) «Программное обеспечение в ВУЗе».

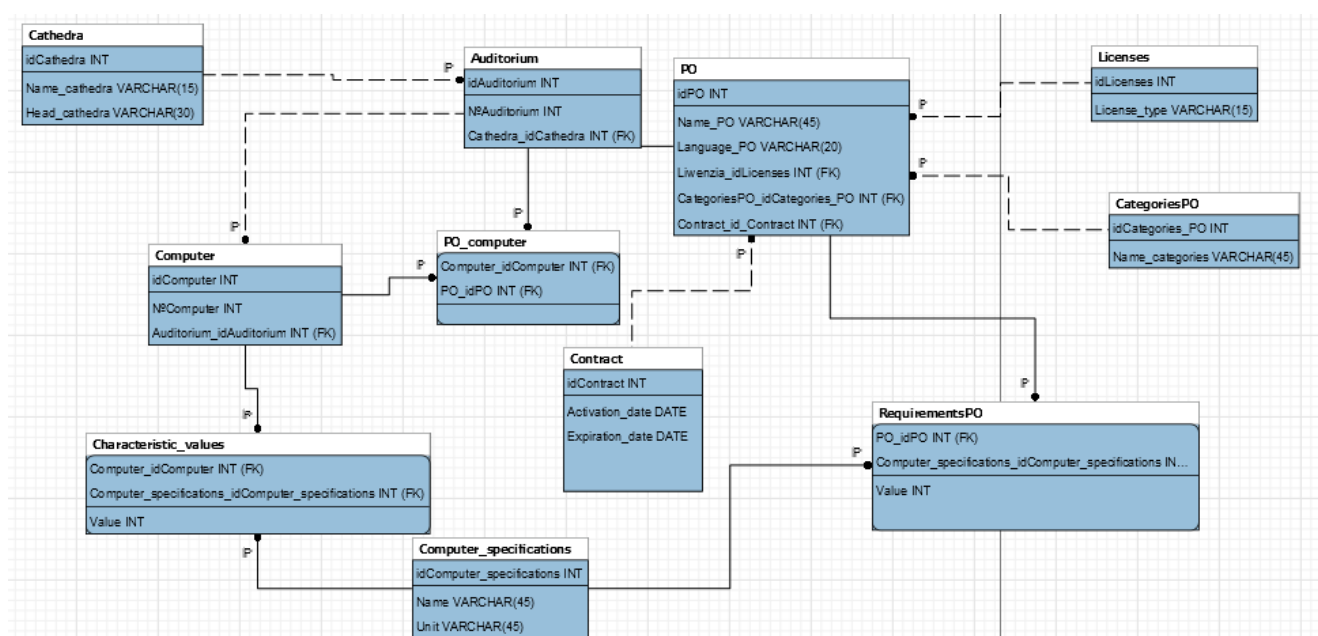


Рисунок 1.1 - Инфологическая модель данных

Сгенерированный sql код в приложении А.

Таблица 1 - Сущности и атрибуты.

Название таблицы	Описание	Атрибуты	Первичный ключ
Cathedra	Кафедра id кафедры Наименование кафедры	idCathedra Name_ Cathedra Head_Cathedra	idCathedra

	Заведующий кафедры		
Auditorium	Аудитория id аудитории № аудитории Внешний ключ	idAuditorium №Auditorium Cathedra_idCathedra	idAuditorium
Computer	Компьютер id компьютера № компьютера Внешний ключ	idComputer №Computer Auditorium_idAuditorium	idComputer
PO_computer	ПО компьютера Внешний ключ Внешний ключ	Computer_idComputer PO_idPO	
PO	ПО id ПО Название ПО Язык ПО Внешний ключ Внешний ключ Внешний ключ	idPO Name_PO Language_PO Licenzia_idLicenses CategoriesPO_idCategorie s_PO Contract_id_Contract	idPO
Licenses	Лицензия id лицензии Тип лицензии	idLicenses License_type	idLicenses
Categories_PO	Категория ПО id категории ПО Название категории ПО	idCategories_PO Name_categories	idCategories_ PO

Contract	Договор id договора Дата активации Дата окончания	idContract Activation_date Expiration_date	idContract
RequirementsPO	Требования ПО Внешний ключ Внешний ключ	PO_idPO Computer_specifications_idComputer_specifications	
Computer_specifications	Технические характеристики компьютера id характеристики компьютера Название Значение	idComputer_specifications Name Unit	idComputer_specifications
Characteristic_values	Значения характеристики Значение Внешний ключ Внешний ключ	Value Computer_idComputer Computer_specifications_idComputer_specifications	

1.2 Связи

Связь Cathedra - Auditorium – неидентифицирующая связь 1:n

Связь Auditorium - Computer – неидентифицирующая связь 1:n

Связь Computer - PO_computer – идентифицирующая связь 1:n

Связь PO_computer - PO – идентифицирующая связь n:1

Связь PO - Licenses – неидентифицирующая связь n:1

Связь PO - Categories_PO – неидентифицирующая связь n:1

Связь PO - Contract – неидентифицирующая связь n:1

Связь PO - RequirementsPO – идентифицирующая связь 1:n

Связь RequirementsPO - Computer_specifications – идентифицирующая связь
n:1

Computer_specifications - Characteristic_values – идентифицирующая связь 1:n

2. Логическое проектирование

2.1 Нормализация

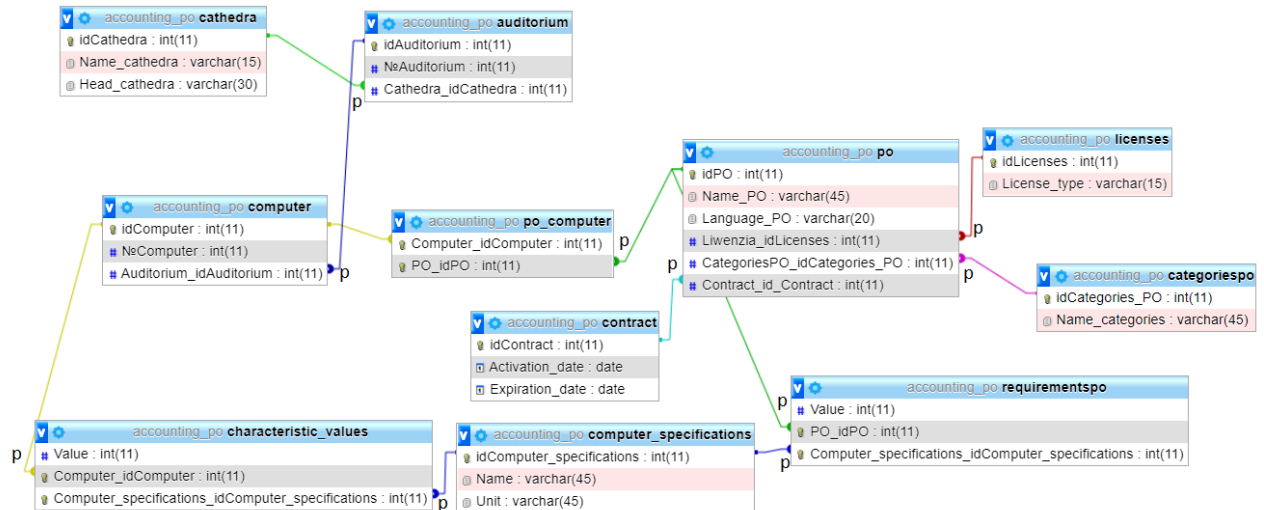


Рисунок 2.1 - Логическая схема

1) Таблица Cathedra:

idCathedra → Name_Cathedra

idCathedra → Head_Cathedra

Соответствует 3 нормальной форме.

2) Таблица Auditorium:

idAuditorium → Name_Cathedra

idAuditorium → Head_Cathedra

idCathedra (таблица Cathedra) → Cathedra_idCathedra

Соответствует 3 нормальной форме.

3) Таблица Computer:

idComputer → №Computer

idComputer → Auditorium_idAuditorium

idAuditorium (таблица Auditorium) → Auditorium_idAuditorium

Соответствует 3 нормальной форме.

4) Таблица PO_computer:

idComputer (таблица Computer) → Computer_idComputer

idPO (таблица PO) → PO_idPO

Соответствует 3 нормальной форме.

5) Таблица PO:

idPO → Name_PO

idPO → Language_PO

idLicenses (таблица Licenses) → Licenzia_idLicenses

idCategoriesPO (таблица CategoriesPO) → CategoriesPO_idCategories_PO

idContract (таблица Contract) → Contract_id_Contract

Соответствует 3 нормальной форме.

6) Таблица Licenses:

idLicenses → License_type

Соответствует 3 нормальной форме.

7) Таблица Contract:

idContract → Activation_date

idContract → Expiration_date

Соответствует 3 нормальной форме.

8) Таблица CategoriesPO:

idCategoriesPO → Name_categories

Соответствует 3 нормальной форме.

9) Таблица RequirementsPO:

idPO (таблица PO) → PO_idPO

idComputer_specifications (таблица Computer_specifications) →

Computer_specifications_idComputer_specifications

PO_idPO, Computer_specifications_idComputer_specifications → Value

Соответствует 3 нормальной форме.

10) Таблица Computer_specifications:

idComputer_specifications → Name

idComputer_specifications → Unit

Соответствует 3 нормальной форме.

11) Таблица Characteristic_values:

idComputer (таблица Computer) → Computer_idComputer

idComputer_specifications (таблица Computer_specifications) →

Computer_specifications_idComputer_specifications

Computer_idComputer, Computer_specifications_idComputer_specifications
→ Value

Соответствует 3 нормальной форме.

2.2 Соответствие нормальным формам

Первая нормальная форма

Основным правилом первой формы является необходимость неделимости значения в каждом поле (столбце) строки – атомарность значений (рис. 2.2).

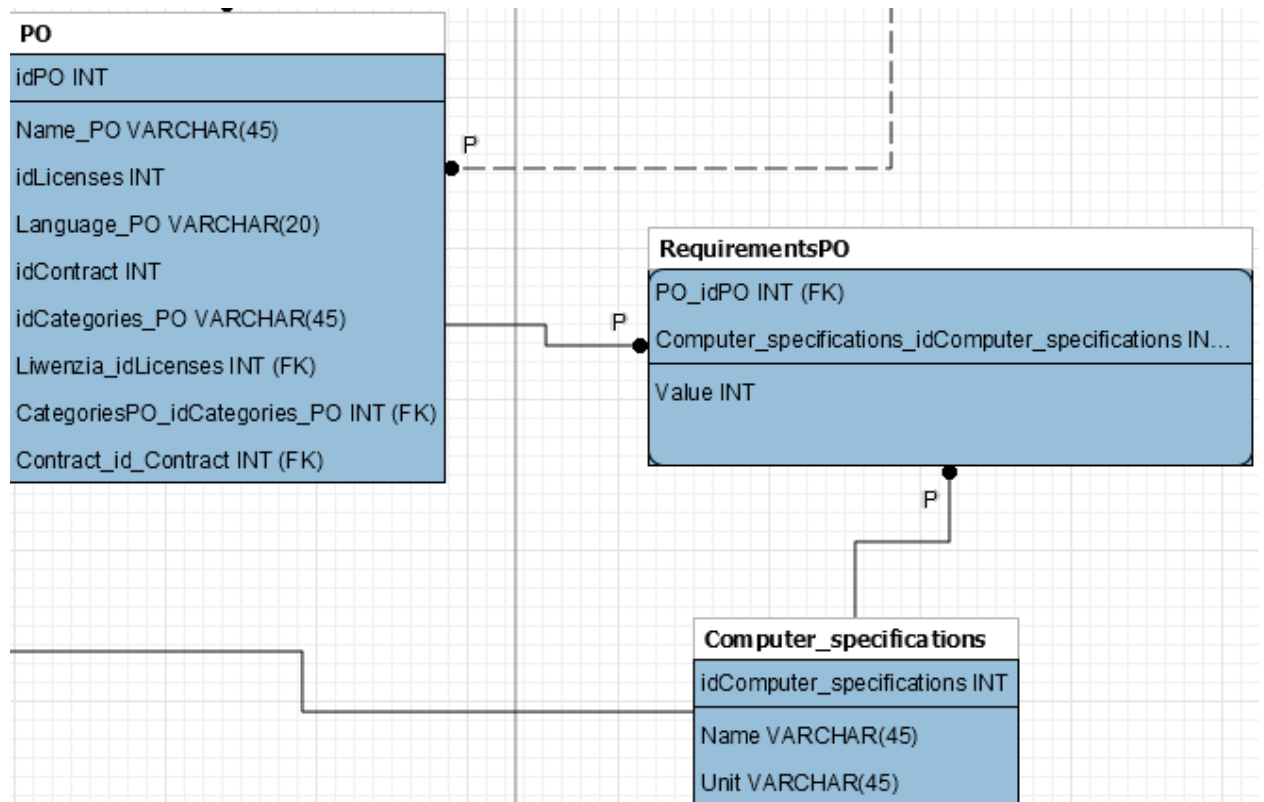


Рисунок 2.2 Первая нормальная форма

Помимо атомарности к первой нормальной форме относятся следующие правила:

- Строки таблиц не должны зависеть друг от друга, т.е. первая запись не должна влиять на вторую и наоборот, вторая на третью и т.д. Размещение записей в таблице не имеет никакого значения.
- Аналогичная ситуация со столбцами записей. Их порядок не должен влиять на понимание информации.
- Каждая строка должна быть уникальна, поэтому для нее определяется первичный ключ, состоящий из одного либо нескольких полей (составной ключ). Первичный ключ не может повторяться в пределах таблицы и служит идентификатором записи.

Вторая нормальная форма

Условием этой формы является отсутствие зависимости неключевых полей от части составного ключа и соответствует первой нормальной форме.

Рассмотрим пример на таблице RequirementsPO(рис 2.3).

RequirementsPO	
PO_idPO INT (FK)	
Computer_specifications_idComputer_specifications IN...	
Value INT	

Рисунок 2.3 Вторая нормальная форма

Атрибут Value зависит от полного составного ключа, а не от его части.

Третья нормальная форма

Условием этой формы является соответствие второй нормальной форме и ни один неключевой атрибут не находится в зависимости ключа.

Рассмотрим пример на таблице PO (рис. 2.4).

PO	
idPO INT	
Name_PO VARCHAR(45)	
idLicenses INT	
Language_PO VARCHAR(20)	
idContract INT	
idCategories_PO VARCHAR(45)	
Liwerzia_idLicenses INT (FK)	
CategoriesPO_idCategories_PO INT (FK)	
Contract_id_Contract INT (FK)	

Рисунок 2.4 Третья нормальная форма

Каждый неключевой атрибут таблицы зависит от первичного ключа IdPO. Нет атрибутов который зависят от других атрибутов.

3. Физическое проектирование

3.1 Типы данных

Для отображения типов данных использовался MySQL.

1) Используемые типы данных в таблице Cathedra изображены на рисунке 3.1.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idCathedra	INT	☒	☒	☒	☐	☐	☐	☒	☐
Name_cathedra	VARCHAR(15)	☐	☒	☐	☐	☐	☐	☐	☐
Head_cathedra	VARCHAR(30)	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.1 - типы данных в таблице Cathedra

2) Используемые типы данных в таблице Auditorium изображены на рисунке 3.2.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idAuditorium	INT	☒	☒	☒	☐	☐	☐	☒	☐
NºAuditorium	INT	☐	☒	☐	☐	☐	☐	☐	☐
Cathedra_idCathedra	INT	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.2 - типы данных в таблице Auditorium

3) Используемые типы данных в таблице Computer изображены на рисунке 3.3.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idComputer	INT	☒	☒	☒	☐	☐	☐	☒	☐
NºComputer	INT	☐	☒	☐	☐	☐	☐	☐	☐
Auditorium_idAuditorium	INT	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.3 - типы данных в таблице Computer

4) Используемые типы данных в таблице PO_computer изображены на рисунке 3.4.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
Computer_idComputer	INT	☒	☒	☐	☐	☐	☐	☐	☐
PO_idPO	INT	☒	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.4 - типы данных в таблице PO_computer

5) Используемые типы данных в таблице PO изображены на рисунке 3.5.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idPO	INT	☒	☒	☒	☐	☐	☐	☒	☐
Name_PO	VARCHAR(45)	☐	☒	☐	☐	☐	☐	☐	☐
Language_PO	VARCHAR(20)	☐	☒	☐	☐	☐	☐	☐	☐
Liwenzia_idLicenses	INT	☐	☒	☐	☐	☐	☐	☐	☐
CategoriesPO_idCategories_...	INT	☐	☒	☐	☐	☐	☐	☐	☐
Contract_id_Contract	INT	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.5 - типы данных в таблице PO

6) Используемые типы данных в таблице Contract изображены на рисунке 3.6.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idContract	INT	☒	☒	☒	☐	☐	☐	☒	☐
Activation_date	DATE	☐	☒	☐	☐	☐	☐	☐	☐
Expiration_date	DATE	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.6 - типы данных в таблице Contract

7) Используемые типы данных в таблице Licenses изображены на рисунке 3.7.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idLicenses	INT	☒	☒	☒	☐	☐	☐	☒	☐
License_type	VARCHAR(15)	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.7 - типы данных в таблице Licenses

8) Используемые типы данных в таблице CategoriesPO изображены на рисунке 3.8.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idCategories_PO	INT	☒	☒	☒	☐	☐	☐	☒	☐
Name_categories	VARCHAR(45)	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.8 - типы данных в таблице CategoriesPO

9) Используемые типы данных в таблице RequirementsPO изображены на рисунке 3.9.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
Value	INT	☐	☒	☐	☐	☐	☐	☐	☐
PO_idPO	INT	☒	☒	☐	☐	☐	☐	☐	☐
Computer_specifications_idC...	INT	☒	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.9 - типы данных в таблице RequirementsPO

10) Используемые типы данных в таблице Computer_specifications изображены на рисунке 3.10.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idComputer_specifications	INT	☒	☒	☒	☐	☐	☐	☒	☐
Name	VARCHAR(45)	☐	☒	☐	☐	☐	☐	☐	☐
Unit	VARCHAR(45)	☐	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.10 - типы данных в таблице Computer_specifications

11) Используемые типы данных в таблице Characteristic_values изображены на рисунке 3.11.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
Value	INT	☐	☒	☐	☐	☐	☐	☐	☐
Computer_idComputer	INT	☒	☒	☐	☐	☐	☐	☐	☐
Computer_specifications_idC...	INT	☒	☒	☐	☐	☐	☐	☐	☐

Рисунок 3.11 - типы данных в таблице Characteristic_values

3.2 Индексы

Для отображения индексов использовался phpMyAdmin.

1) Индексы в таблице Cathedra изображены на рисунке 3.12

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idCathedra	3	A	Нет	
idКафедры_UNIQUE	BTREE	Да	Нет	idCathedra	3	A	Нет	

Рисунок 3.12 - индексы в таблице Cathedra

PRIMARY – первичный ключ столбец idCathedra

idКафедры_UNIQUE – уникальность первичного ключа

2) Индексы в таблице Auditorium изображены на рисунке 3.13

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idAuditorium	8	A	Нет	
idAuditorium_UNIQUE	BTREE	Да	Нет	idAuditorium	8	A	Нет	
fk_Ayditoria_Cathedra1_idx	BTREE	Нет	Нет	Cathedra_idCathedra	8	A	Нет	

Рисунок 3.13 - индексы в таблице Auditorium

PRIMARY – первичный ключ столбец idAuditorium

idAuditorium_UNIQUE - уникальность первичного ключа

fk_Ayditoria_Cathedra1_idx – внешний ключ, передаваемый из таблицы Cathedra

3) Индексы в таблице Computer изображены на рисунке 3.14

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idComputer	29	A	Нет	
idComputer_UNIQUE	BTREE	Да	Нет	idComputer	29	A	Нет	
fk_Computer_Auditorium1_idx	BTREE	Нет	Нет	Auditorium_idAuditorium	29	A	Нет	

Рисунок 3.14 - индексы в таблице Computer

PRIMARY – первичный ключ столбец idComputer

idComputer_UNIQUE - уникальность первичного ключа

fk_Computer_Auditorium1_idx – внешний ключ, передаваемый из таблицы Auditorium

4) Индексы в таблице PO_computer изображены на рисунке 3.15

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	Computer_idComputer	99	A	Нет	
				PO_idPO	99	A	Нет	
fk_PO_computer_Computer1_idx	BTREE	Нет	Нет	Computer_idComputer	99	A	Нет	
fk_PO_computer_PO1_idx	BTREE	Нет	Нет	PO_idPO	14	A	Нет	

Рисунок 3.15 - индексы в таблице PO_computer

PRIMARY – первичный ключ столбец Computer_idComputer, PO_idPO

fk_PO_computer_Computer1_idx - внешний ключ, передаваемый из таблицы Computer

fk_Computer_Auditorium1_idx – внешний ключ, передаваемый из таблицы ро

5) Индексы в таблице ро изображены на рисунке 3.16

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idPO	7	A	Нет	
idPO_UNIQUE	BTREE	Да	Нет	idPO	7	A	Нет	
fk_PO_Liwenzia1_idx	BTREE	Нет	Нет	Liwenzia_idLicenses	7	A	Нет	
fk_PO_CategoriesPO1_idx	BTREE	Нет	Нет	CategoriesPO_idCategories_PO	7	A	Нет	
fk_PO_Contract1_idx	BTREE	Нет	Нет	Contract_id_Contract	7	A	Нет	

Рисунок 3.16 - индексы в таблице ро

PRIMARY – первичный ключ столбец idPO

idPO_UNIQUE - уникальность первичного ключа

fk_PO_Liwenzia1_idx - внешний ключ, передаваемый из таблицы licenses

fk_PO_CategoriesPO1_idx – внешний ключ, передаваемый из таблицы CategoriesPO

fk_PO_Contract1_idx - внешний ключ, передаваемый из таблицы Contract

6) Индексы в таблице Contract изображены на рисунке 3.17

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idContract	7	A	Нет	
№Договора_UNIQUE	BTREE	Да	Нет	idContract	7	A	Нет	

Рисунок 3.17 - индексы в таблице Contract

PRIMARY – первичный ключ столбец idContract

№Договора_UNIQUE - уникальность первичного ключа

7) Индексы в таблице Licenses изображены на рисунке 3.18

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idLicenses	4	A	Нет	
idЛицензии_UNIQUE	BTREE	Да	Нет	idLicenses	4	A	Нет	

Рисунок 3.18 - индексы в таблице Licenses

PRIMARY – первичный ключ столбец idLicenses

№Лицензии_UNIQUE - уникальность первичного ключа

8) Индексы в таблице CategoriesPO изображены на рисунке 3.19

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idCategories_PO	4	A	Нет	
idКатегории ПО_UNIQUE	BTREE	Да	Нет	idCategories_PO	4	A	Нет	

Рисунок 3.19 - индексы в таблице CategoriesPO

PRIMARY – первичный ключ столбец idCategories_PO

№Категории_UNIQUE - уникальность первичного ключа

9) Индексы в таблице RequirementsPO изображены на рисунке 3.20

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	PO_idPO	20	A	Нет	
				Computer_specifications_idComputer_specifications	20	A	Нет	
fk_RequirementsPO_PO1_idx	BTREE	Нет	Нет	PO_idPO	20	A	Нет	
fk_RequirementsPO_Computer_specifications1_idx	BTREE	Нет	Нет	Computer_specifications_idComputer_specifications	6	A	Нет	

Рисунок 3.20 - индексы в таблице RequirementsPO

PRIMARY – первичный ключ столбец PO_idPO,

Computer_specifications_idComputer_specifications

fk_RequirementsPO_PO1_idx - внешний ключ, передаваемый из таблицы PO

fk_RequirementsPO_Computer_specifications1_idx - внешний ключ,

передаваемый из таблицы Computer_specifications

10) Индексы в таблице Computer_specifications изображены на рисунке 3.21

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	idComputer_specifications	2	A	Нет	
idХарактеристикиКомпьютера_UNIQUE	BTREE	Да	Нет	idComputer_specifications	2	A	Нет	

Рисунок 3.21 - индексы в таблице Computer_specifications

PRIMARY – первичный ключ столбец idComputer_specifications

№ХарактеристикиКомпьютера_UNIQUE - уникальность первичного

ключа

11) Индексы в таблице Characteristic_values изображены на рисунке 3.22

Имя индекса	Тип	Уникальный	Упакован	Столбец	Уникальных элементов	Сравнение	Null	Комментарий
PRIMARY	BTREE	Да	Нет	Computer_idComputer	81	A	Нет	
				Computer_specifications_idComputer_specifications	81	A	Нет	
fk_Characteristic_values_Computer1_idx	BTREE	Нет	Нет	Computer_idComputer	81	A	Нет	
fk_Characteristic_values_Computer_specifications1_idx	BTREE	Нет	Нет	Computer_specifications_idComputer_specifications	6	A	Нет	

Рисунок 3.22 - индексы в таблице Characteristic_values

PRIMARY – первичный ключ столбец Computer_idComputer,

Computer_specifications_idComputer_specifications

fk_Characteristic_values_Computer1_idx - внешний ключ, передаваемый из

таблицы Computer

fk_Characteristic_values_Computer_specifications1_idx - внешний ключ,

передаваемый из таблицы Computer_specifications

4 Разработка структуры приложения.

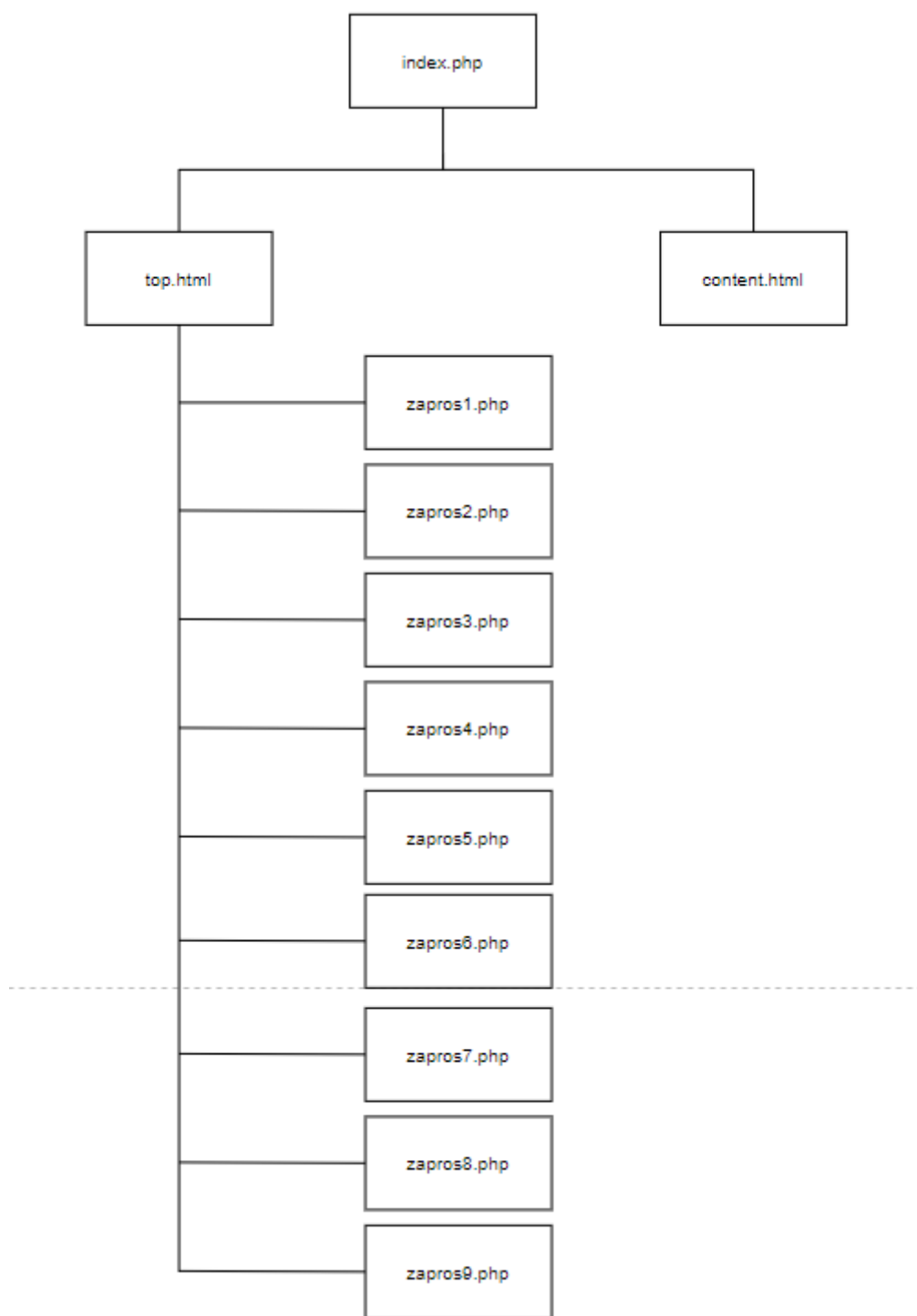


Рисунок 4 – структура приложения

index.php – основной файл вызывающий Top.html и Content.html.

top.html – файл вызывающий zapros1.php, zapros2.php, zapros3.php, zapros4.php, zapros5.php, zapros6.php, zapros7.php, zapros8.php, zapros9.php по нажатию на соответствующую кнопку.

content.html – сюда происходит вывод файлов zapros1.php, zapros2.php, zapros3.php, zapros4.php, zapros5.php, zapros6.php, zapros7.php, zapros8.php, zapros9.php.

zapros1.php – файл содержащий запрос - ПО написанное на языке Java.

zapros2.php – файл содержащий запрос - компьютеры подходящие под Altium по оперативной памяти.

zapros3.php – файл содержащий запрос - количество компьютеров с Altium по кафедрам.

zapros4.php – файл содержащий запрос - обновление договора истекающего в 2020г.

zapros5.php – файл содержащий запрос - договор.

zapros6.php – файл содержащий запрос - добавление данных.

zapros7.php – файл содержащий запрос - нахождение min/max для занимаемой памяти по категории ПО.

zapros8.php – файл содержащий запрос - программные средства по аудиториям.

zapros9.php – файл содержащий запрос - ПО.

5 Разработка приложения

5.1 Разработка запросов

1) ПО написанное на языке Java.

Код запроса:

```
SELECT `po`.`Name_PO`, `po`.`Language_PO`  
FROM `po`  
WHERE `po`.`Language_PO` = 'Java'
```

Результат работы запроса представлен на рис. 5.1.

Название ПО	Язык ПО
Altium	Java
WinRAR	Java

Рисунок 5.1 – результат работы запроса

2) Компьютеры подходящие под Altium по оперативной памяти.

Код запроса:

```
SELECT `computer`.`idComputer`, `po`.`Name_PO`,  
`computer_specifications`.`Name`, `characteristic_values`.`Value`  
FROM `computer`, `po`  
    INNER JOIN `requirementspo` ON `requirementspo`.`PO_idPO` = `po`.`idPO`  
    INNER JOIN `computer_specifications` ON  
`requirementspo`.`Computer_specifications_idComputer_specifications` =  
`computer_specifications`.`idComputer_specifications`  
    INNER JOIN `characteristic_values` ON  
`characteristic_values`.`Computer_specifications_idComputer_specifications` =  
`computer_specifications`.`idComputer_specifications`  
WHERE `po`.`Name_PO` = 'Altium' AND `computer_specifications`.`Name` =  
'ОперативнаяПамять' AND `characteristic_values`.`Value` >=  
`requirementspo`.`Value`  
GROUP BY idComputer
```

Результат работы запроса представлен на рис. 5.2.

id компьютера	Название ПО	Название характеристики	Значение
1	Altium	ОперативнаяПамять	12
2	Altium	ОперативнаяПамять	12
3	Altium	ОперативнаяПамять	12
4	Altium	ОперативнаяПамять	12
5	Altium	ОперативнаяПамять	12
6	Altium	ОперативнаяПамять	12
7	Altium	ОперативнаяПамять	12
8	Altium	ОперативнаяПамять	12
9	Altium	ОперативнаяПамять	12
10	Altium	ОперативнаяПамять	12
11	Altium	ОперативнаяПамять	12
12	Altium	ОперативнаяПамять	12
13	Altium	ОперативнаяПамять	12
14	Altium	ОперативнаяПамять	12
15	Altium	ОперативнаяПамять	12
16	Altium	ОперативнаяПамять	12
17	Altium	ОперативнаяПамять	12
18	Altium	ОперативнаяПамять	12

Рисунок 5.2 – результат работы запроса

3) Количество компьютеров с Altium по кафедрам.

Код запроса:

```
SELECT `cathedra`.`Name_cathedra`,`po`.`Name_PO`,COUNT(*)
```

```
FROM `cathedra`
```

```
    INNER JOIN `auditorium` ON `auditorium`.`Cathedra_idCathedra` =
`cathedra`.`idCathedra`
```

```
    INNER JOIN `computer` ON `computer`.`Auditorium_idAuditorium` =
`auditorium`.`idAuditorium`
```

```
INNER JOIN `po_computer` ON `po_computer`.`Computer_idComputer` =
`computer`.`idComputer`
```

```
INNER JOIN `po` ON `po_computer`.`PO_idPO` = `po`.`idPO`
```

```
WHERE `po`.`Name_PO` = 'Altium'
```

```
GROUP BY Name_cathedra
```

Результат работы запроса представлен на рис. 5.3.

Название кафедры	Название ПО	Количество
Инженерное прое	Altium	5
Информационная	Altium	3
САПР	Altium	5

Рисунок 5.3 – результат работы запроса

4) Обновление договора истекающего в 2020г.

Код запроса:

```
UPDATE `contract` SET `contract`.`Expiration_date` =
```

```
DATE_ADD(`Expiration_date`, INTERVAL 1 YEAR)
```

```
WHERE (((Year(`contract`.`Expiration_date`))=2020))
```

Результат работы запроса представлен на рис. 5.4.

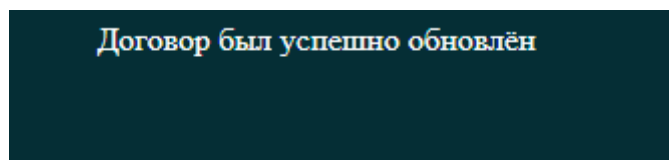


Рисунок 5.4 – результат работы запроса

5) Вывод Договора.

Код запроса:

```
SELECT * FROM `contract`
```

Результат работы запроса представлен на рис. 5.5.

id договора	Дата активации	Дата окончания
1	2015-11-02	2021-11-03
2	2017-06-12	2026-06-12
3	2018-09-27	2021-09-28
4	2020-10-06	2021-10-06
5	2019-04-12	2021-12-15
6	2016-05-23	2023-05-23
7	2020-09-02	2010-09-02

Рисунок 5.5 – результат работы запроса

6) Добавление данных.

Код запроса:

```
INSERT INTO `po`
(`Name_PO`,`Language_PO`,`Liwenzia_idLicenses`,`CategoriesPO_idCategories_PO`,`Contract_id_Contract`) VALUES
('$Name_PO','$Language_PO','$Liwenzia_idLicenses','$CategoriesPO_idCategories_PO','$Contract_id_Contract')
```

Результат работы запроса представлен на рис. 5.6.

Ведите данные для добавления

Название

Язык ПО

id Лицензии

id Категории ПО

id Договора

Добавить

Рисунок 5.6 – результат работы запроса

7) Нахождение min/max для занимаемой памяти по категории ПО.

Код запроса:

```
SELECT `categoriespo`.`Name_categories`,MAX(VALUE),MIN(VALUE)
FROM `categoriespo`
INNER JOIN `po` ON
`po`.`CategoriesPO_idCategories_PO`=`categoriespo`.`idCategories_PO`
INNER JOIN `requirementspo` ON `requirementspo`.`PO_idPO` = `po`.`idPO`
```

```
WHERE `requirementspo`.`Computer_specifications_idComputer_specifications`
= 3
GROUP BY Name_categories
```

Результат работы запроса представлен на рис. 5.7.

Название категории	Max,GB	Min,GB
Архиватор	35	35
ГрафическийРедактор	90	55
МатематическоеПО	40	40
ОперационнаяСистема	60	50
ОфиснаяПрограмма	20	20

Рисунок 5.7 – результат работы запроса

8) Программные средства по аудиториям.

Код запроса:

```
SELECT `auditorium`.`№Auditorium`, `po`.`Name_PO`
FROM `auditorium`, `po` ORDER BY `auditorium`.`№Auditorium` ASC
```

Результат работы запроса представлен на рис. 5.8.

№ аудитории	Название ПО
102	Blender
102	MATLAB
102	MicrosoftWord
102	WinRAR
102	MicrosoftWindows
102	Linux
102	Altium
103	MicrosoftWindows
103	Linux
103	Altium
103	Blender
103	MATLAB
103	MicrosoftWord
103	WinRAR
106	Linux
106	Altium
106	Blender
106	MATLAB

Рисунок 5.8 – результат работы запроса

9) Вывод ПО.

Код запроса:

SELECT * FROM `po`

Результат работы запроса представлен на рис. 5.9.

id ПО	Имя ПО	Язык ПО	id Лицензии	id Категории ПО	id Договора
1	MicrosoftWindows	C++	5	1	1
2	Linux	C++	4	1	2
3	Altium	Java	5	2	3
4	Blender	C++	1	2	4
5	MATLAB	C#	1	3	5
6	MicrosoftWord	C#	5	4	6
7	WinRAR	Java	1	5	7

Рисунок 5.9 – результат работы запроса

5.2 Разработка программных модулей

1) Файл index.php

С помощью `frameset` создаём 2 фрейма.

С помощью `rows` указываем в % количество поля для каждого фрейма.

С помощью `frame src` выводим файлы-параметры `content.html` и `top.html` для фреймов. Задаём параметры полосы прокрутки и возможности изменения размеров фреймов.

2) Файл content.html

Параметры для фрейма, выводящего результат.

В блоке `<body>` задаём цвет фона.

С помощью `img src` выводим картинку и задаём её размерности на главный экран.

3) Файл top.html

В блоке `<style>` задаём стили для текста, кнопок.

В блоке `<body>` задаётся создание кнопок и изменение цвета фона.

Блок `<body>`:

С помощью `form action` подключаем к кнопкам наши `php` файлы с запросами, указываем вывод результата в `content.html`.

С помощью `button` создаём кнопки

4) Файл zapros3.php.

В блоке `<style>` мы задаём параметры для изменения внешнего вида таблиц и текста.

В блоке `<body>` задаётся `php` код для запроса и изменение цвета фона.

Блок `<body>`:

`$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');` - подключение нашей базы данных к сайту.

`$query` – сюда записываем наш `sql` код запроса

if (\$result = mysqli_query(\$connect,\$query)){ - если результат входит в нашу базу данных.

if (\$result->num_rows) { - определение кол-ва строк.

С помощью echo "<table>" открываем блок конструктора таблицы.

С помощью echo добавляем новые столбцы с нужным наименованием.

Через while, echo и \$row['Название столбца'] обращаемся к строкам и заполняем таблицу с условием нашего запроса.

5) Файл zapros6.php.

В блоке <style> мы задаём параметры для изменения внешнего вида таблиц и текста.

В блоке <body> задаётся php код для запроса и изменение цвета фона.

Блок <body>:

С помощью form задаём поля для ввода данных.

\$connect = mysqli_connect('localhost', 'root', '', 'accounting_po'); - подключение нашей базы данных к сайту.

\$query = "SELECT `po`.`\*` FROM po"; - выбор таблицы, куда будут заноситься данные.

С помощью POST указываем в какие столбцы заносятся данные.

С помощью mysqli_query(\$connect, query: - подключаемся к базе данных и выполняем запрос на добавление данных.

5.3 Интерфейс работы с приложением

1) Главная страница сайта изображена на рис. 5.10.

На главную страницу можно попасть нажав на кнопку в виде дома.

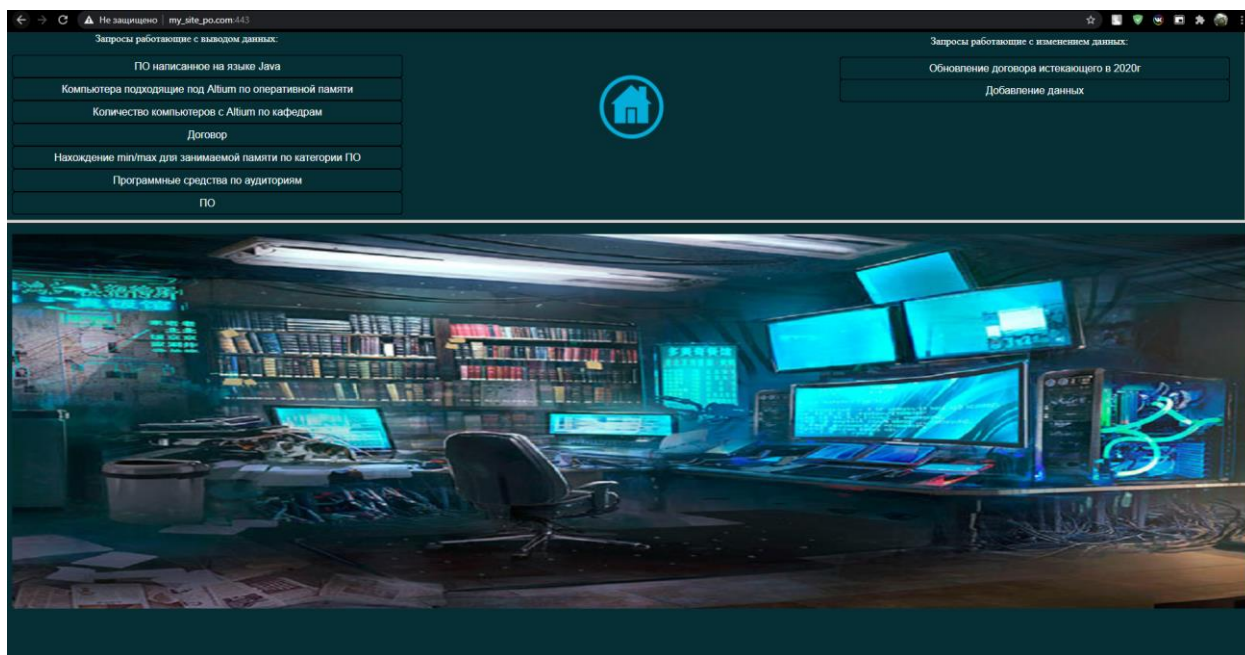


Рисунок 5.10 – главная страница

2) Результат нажатия на кнопку – ПО написанное на языке Java – представлен на рис. 5.11.

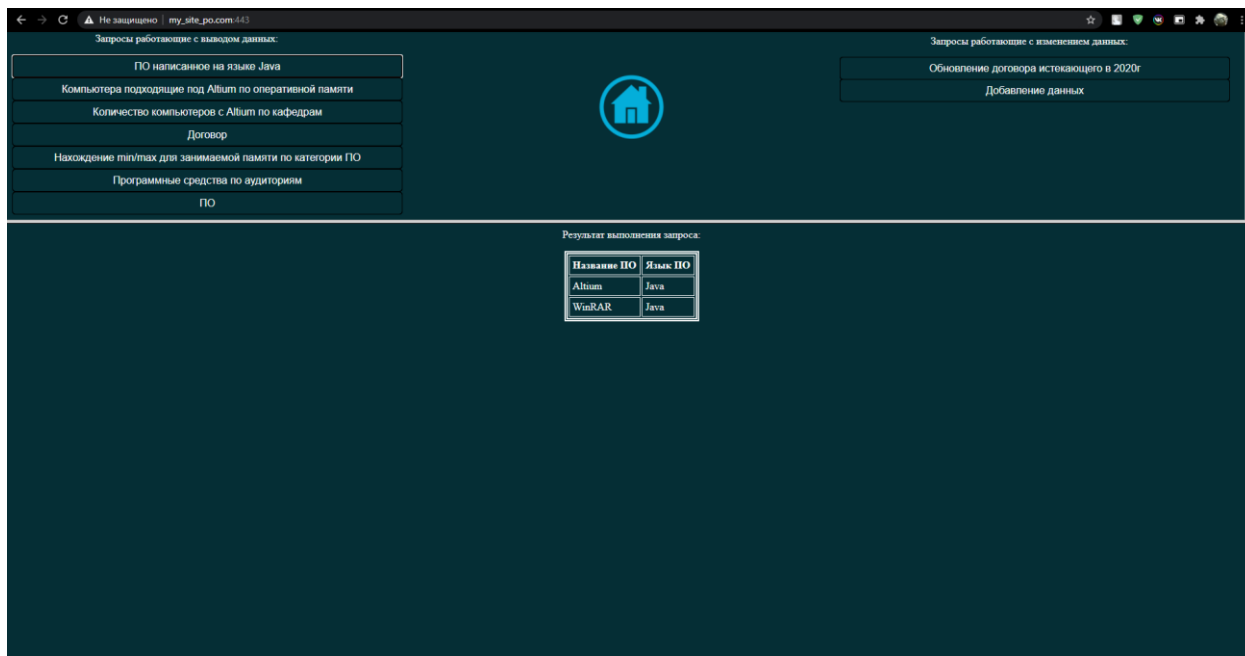
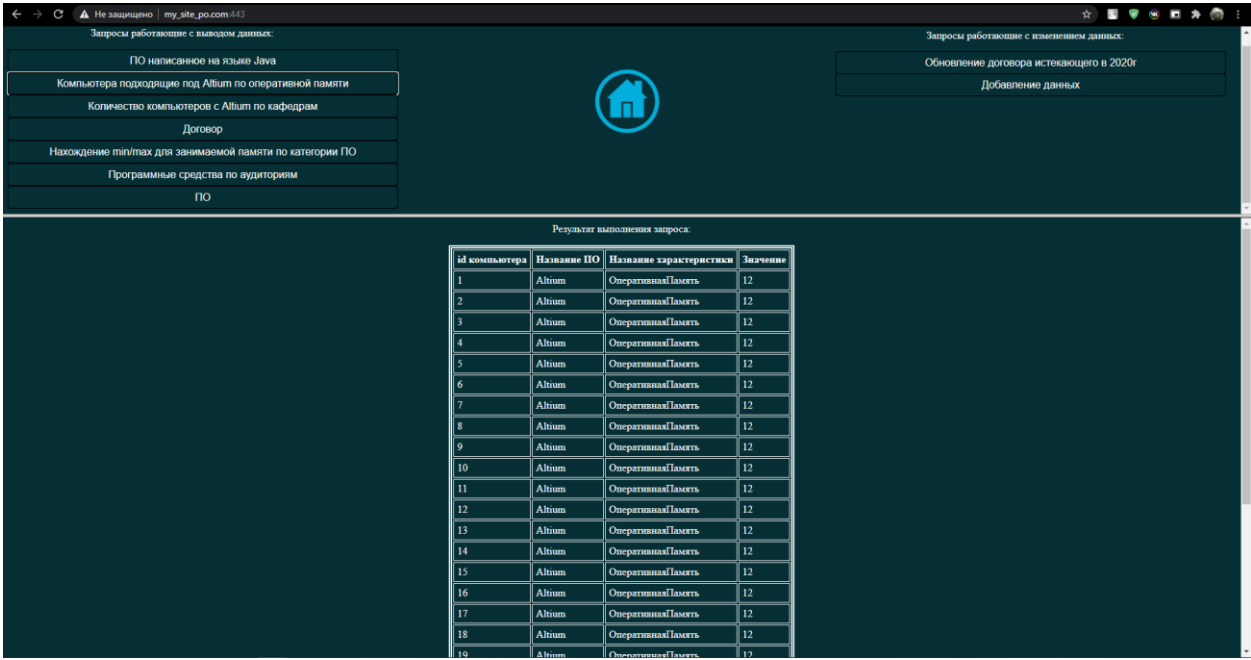


Рисунок 5.11 – результат работы запроса

3) Результат нажатия на кнопку – Компьютеры подходящие под Altium по оперативной памяти – представлен на рис. 5.12.



Запросы работающие с выводом данных:

- ПО написанное на языке Java
- Компьютеры подходящие под Altium по оперативной памяти
- Количество компьютеров с Altium по кафедрам
- Договор
- Нахождение min/max для занимаемой памяти по категории ПО
- Программные средства по аудиториям
- ПО

Запросы работающие с изменением данных:

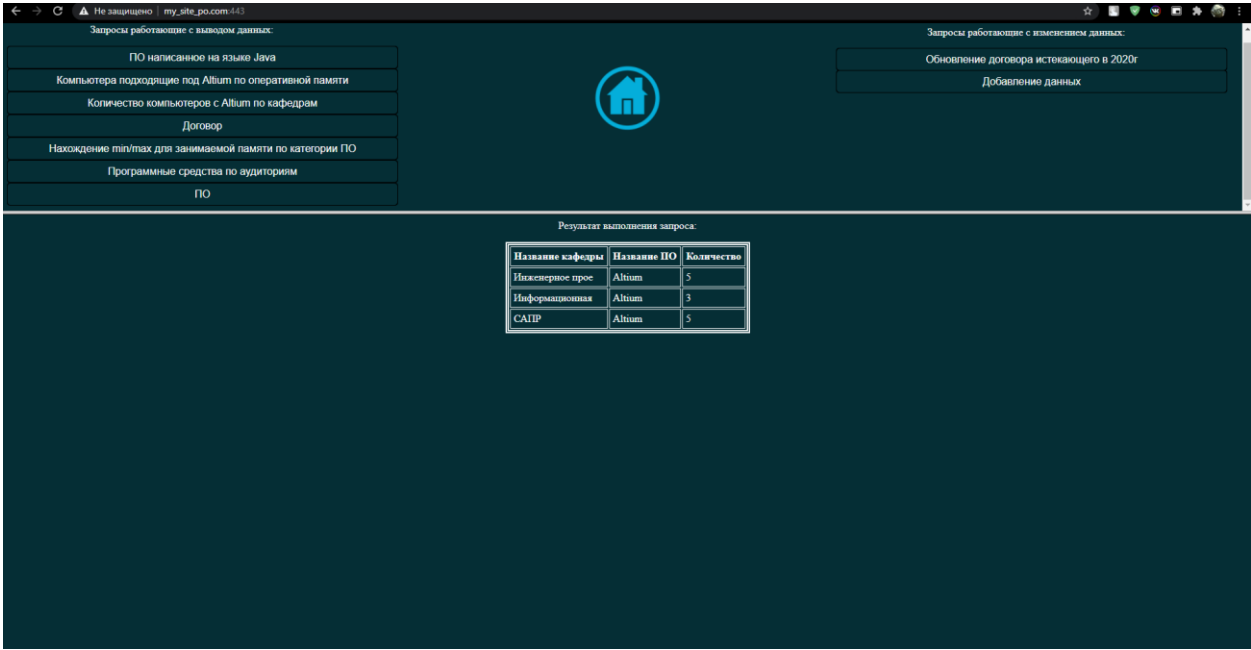
- Обновление договора истекающего в 2020г
- Добавление данных

Результат выполнения запроса:

id компьютера	Название ПО	Название характеристики	Значение
1	Altium	Оперативная Память	12
2	Altium	Оперативная Память	12
3	Altium	Оперативная Память	12
4	Altium	Оперативная Память	12
5	Altium	Оперативная Память	12
6	Altium	Оперативная Память	12
7	Altium	Оперативная Память	12
8	Altium	Оперативная Память	12
9	Altium	Оперативная Память	12
10	Altium	Оперативная Память	12
11	Altium	Оперативная Память	12
12	Altium	Оперативная Память	12
13	Altium	Оперативная Память	12
14	Altium	Оперативная Память	12
15	Altium	Оперативная Память	12
16	Altium	Оперативная Память	12
17	Altium	Оперативная Память	12
18	Altium	Оперативная Память	12
19	Altium	Оперативная Память	12

Рисунок 5.12 – результат работы запроса

4) Результат нажатия на кнопку – Количество компьютеров с Altium по кафедрам – представлен на рис. 5.13.



Запросы работающие с выводом данных:

- ПО написанное на языке Java
- Компьютеры подходящие под Altium по оперативной памяти
- Количество компьютеров с Altium по кафедрам
- Договор
- Нахождение min/max для занимаемой памяти по категории ПО
- Программные средства по аудиториям
- ПО

Запросы работающие с изменением данных:

- Обновление договора истекающего в 2020г
- Добавление данных

Результат выполнения запроса:

Название кафедры	Название ПО	Количество
Инженерное прое	Altium	5
Информационная	Altium	3
САПР	Altium	5

Рисунок 5.13 – результат работы запроса

5) Результат нажатия на кнопку – Договор – представлен на рис. 5.14.

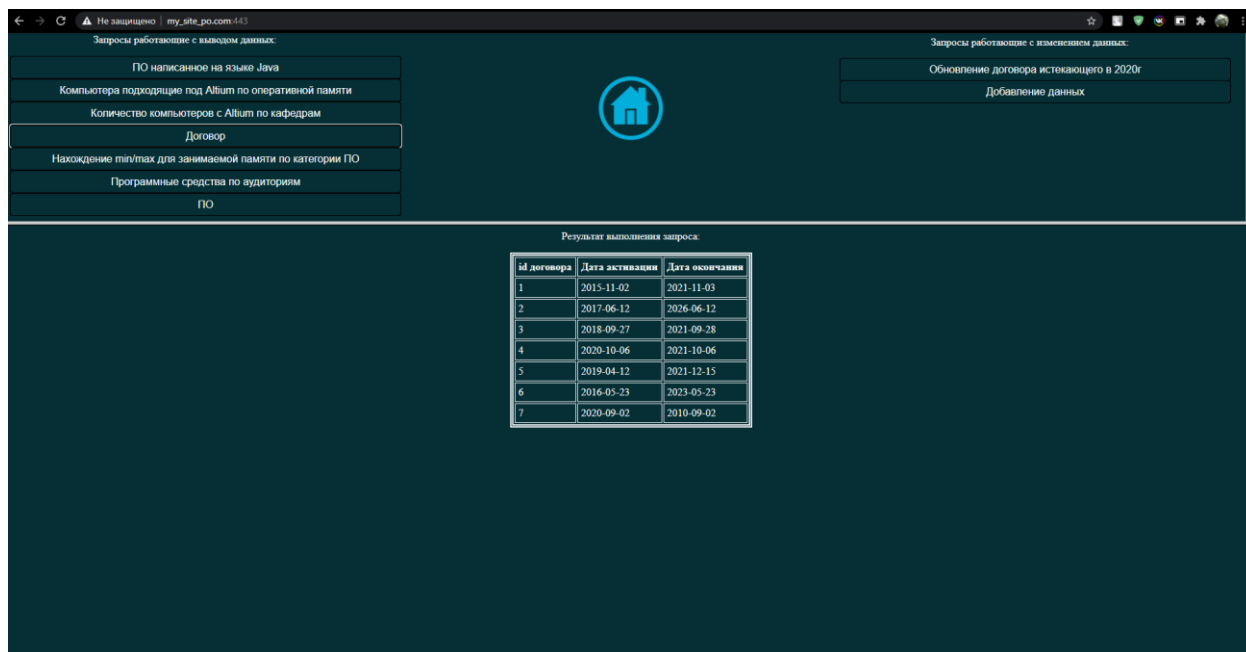


Рисунок 5.14 – результат работы запроса

6) Результат нажатия на кнопку – Нахождение min/max для занимаемой памяти по категории ПО – представлен на рис. 5.15.

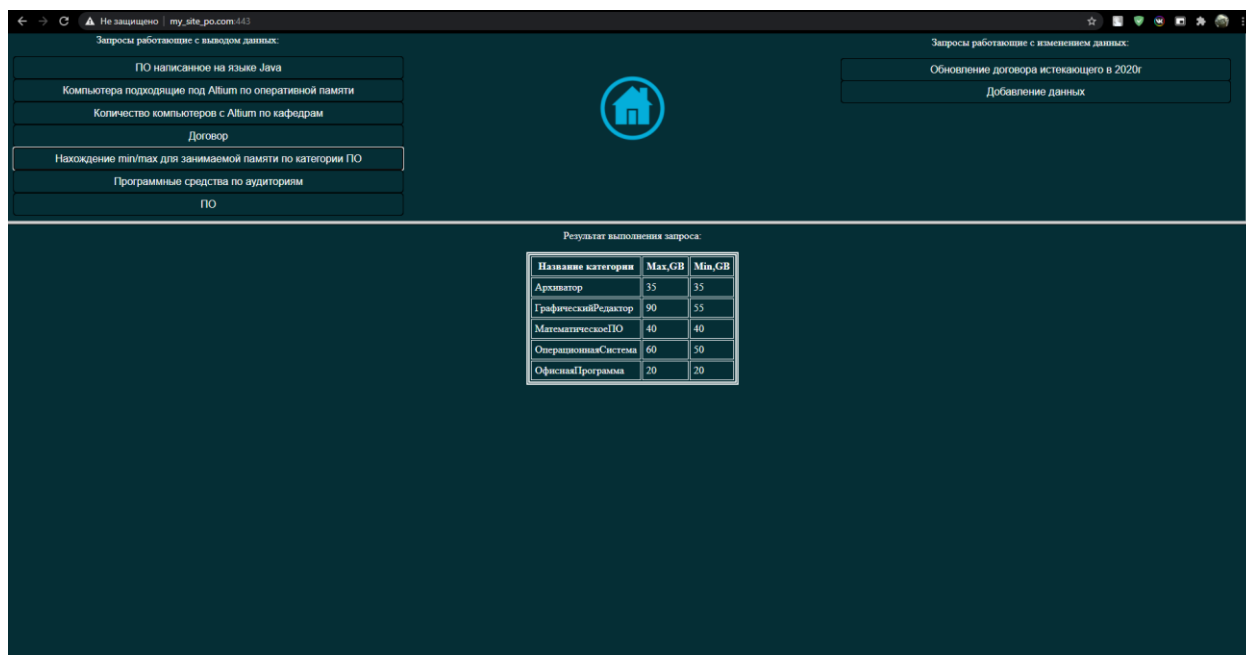


Рисунок 5.15 – результат работы запроса

7) Результат нажатия на кнопку – Программные средства по аудиториям – представлен на рис. 5.16.

Запросы работающие с выводом данных:

- ПО написанное на языке Java
- Компьютера подходящие под Altium по оперативной памяти
- Количество компьютеров с Altium по кафедрам
- Договор
- Нахождение min/max для занимаемой памяти по категории ПО
- Программные средства по аудиториям**
- ПО

Запросы работающие с изменением данных:

- Обновление договора истекающего в 2020г
- Добавление данных

Результат выполнения запроса:

№ аудитории	Название ПО
102	Blender
102	MATLAB
102	MicrosoftWord
102	WinRAR
102	MicrosoftWindows
102	Linux
102	Altium
103	MicrosoftWindows
103	Linux
103	Altium
103	Blender
103	MATLAB
103	MicrosoftWord
103	WinRAR
106	Linux
106	Altium
106	Blender
106	MATLAB
106	MicrosoftWord

Рисунок 5.16 – результат работы запроса

8) Результат нажатия на кнопку – ПО – представлен на рис. 5.17.

Запросы работающие с выводом данных:

- ПО написанное на языке Java
- Компьютера подходящие под Altium по оперативной памяти
- Количество компьютеров с Altium по кафедрам
- Договор
- Нахождение min/max для занимаемой памяти по категории ПО
- Программные средства по аудиториям
- ПО**

Запросы работающие с изменением данных:

- Обновление договора истекающего в 2020г
- Добавление данных

Результат выполнения запроса:

id ПО	Имя ПО	Язык ПО	id Лицензия	id Категории ПО	id Договора
1	MicrosoftWindows	C++	5	1	1
2	Linux	C++	4	1	2
3	Altium	Java	5	2	3
4	Blender	C++	1	2	4
5	MATLAB	C#	1	3	5
6	MicrosoftWord	C#	5	4	6
7	WinRAR	Java	1	5	7

Рисунок 5.17 – результат работы запроса

9) Результат нажатия на кнопку – Обновление договора истекающего в 2020г – представлен на рис. 5.18.

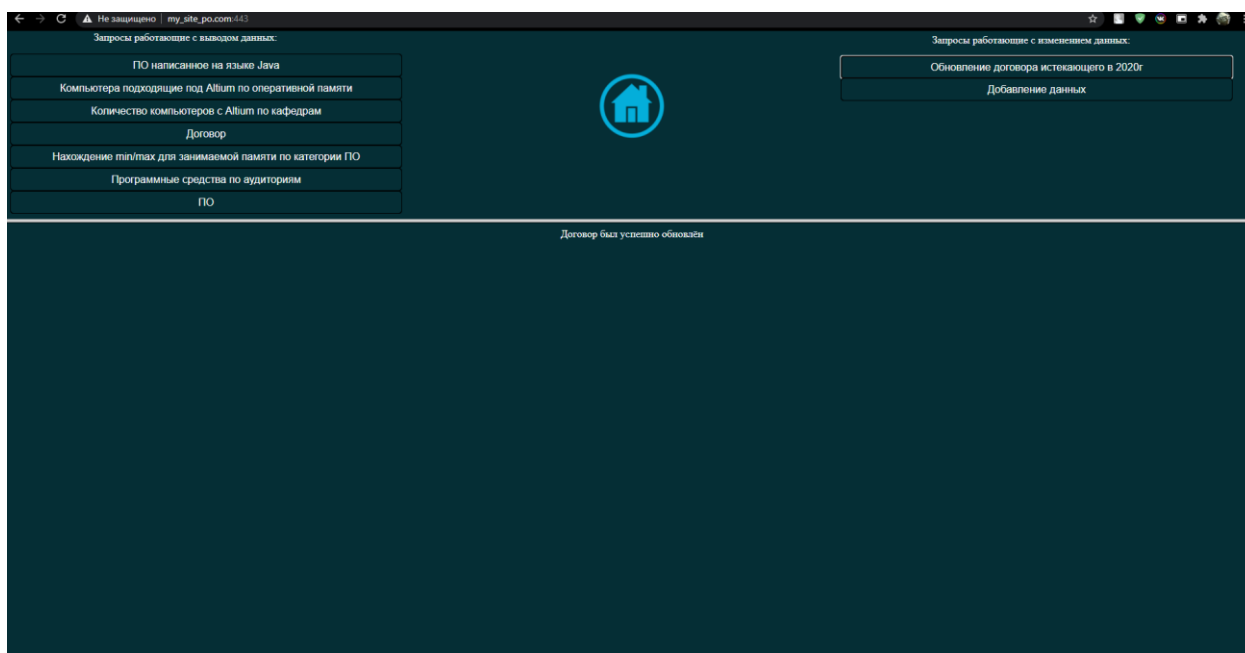


Рисунок 5.18 – результат работы запроса

10) Результат нажатия на кнопку – Добавление данных – представлен на рис. 5.19.

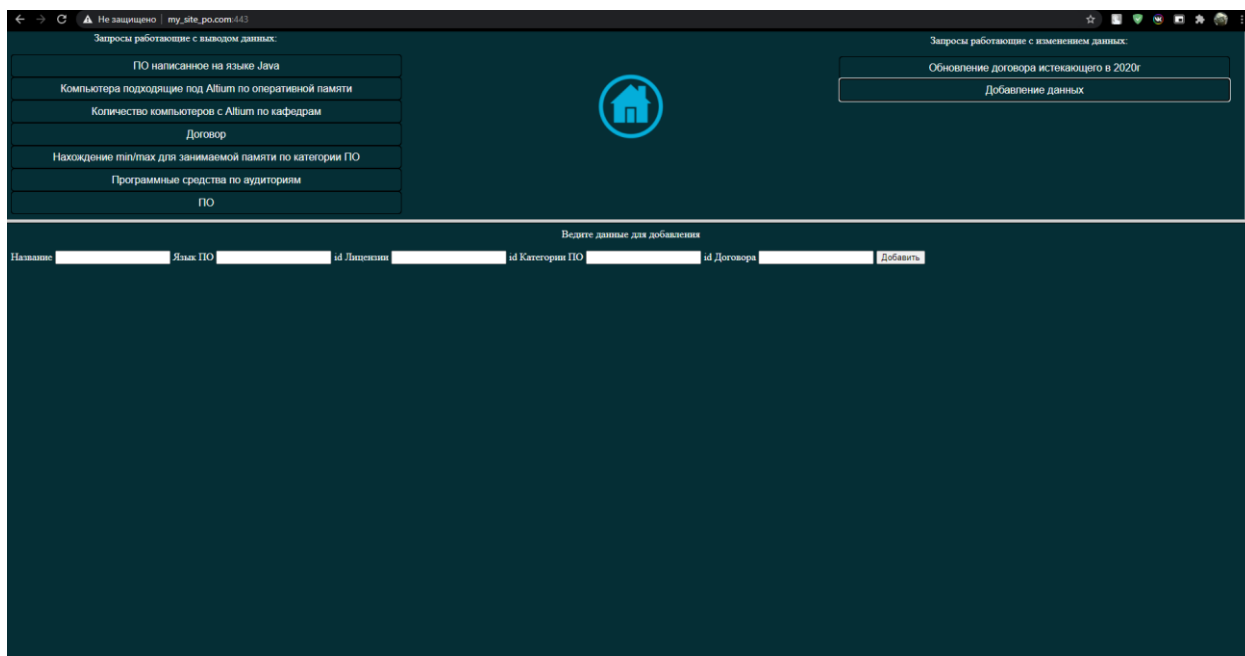


Рисунок 5.19 – результат работы запроса

Заключение

В данной курсовой работе была спроектирована база данных и разработано web приложение обслуживания БД.

Инфологическое проектирование было выполнено при помощи CASE-средств. На этапе логического проектирования была выполнена проверка на соответствие нормальным формам. Разработанное приложение выполняет функции ввода, добавления, поиск данных по ключу. Были использованы операторы языка SQL. Запросы к БД были оптимизированы. При вводе и изменении данных выполняется проверка на целостность данных.

Список используемой литературы

1. Максим Кузнецов, Игорь Симдянов - Самоучитель PHP 2018.
2. Мэтт Зандстра PHP. Объекты, шаблоны и методики программирования 2011.
3. Алан Бьюли Изучаем SQL 2007.
4. Крис Фиайли SQL 2013.
5. Алекс Кригель и др. SQL. Библия пользователя, 2-е издание 2010.
6. Tom Butler & Kevin Yank PHP & MySQL Novice to Ninja 2017.
7. Линн Бейли, Майкл Моррисон Изучаем PHP и MySQL 2010.
8. Joel Murach & Ray Harris Murach's PHP and MySQL 2010.
9. Alan Forbes The Joy of PHP Programming 2015.
10. Маклафлин Бретт PHP и MySQL. Исчерпывающее руководство 2014.
11. Робин Никсон Создаем динамические веб-сайты с помощью PHP, MySQL 2013.
12. Laura Thompson and Luke Welling PHP and MySQL Web Development 2016.
13. Cyberpunk 2077 URL:https://ru.wikipedia.org/wiki/Cyberpunk_2077 (дата обращения: 20.12.2020).

Приложение А

Генерация SQL кода

-- MySQL Script generated by MySQL Workbench

-- Wed Dec 23 18:11:05 2020

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,

UNIQUE_CHECKS=0;

SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,

FOREIGN_KEY_CHECKS=0;

SET @OLD_SQL_MODE=@@SQL_MODE,

SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----

-- Schema Accounting_PO

-- -----

-- -----

-- Schema Accounting_PO

-- -----

CREATE SCHEMA IF NOT EXISTS `Accounting\_PO` DEFAULT

CHARACTER SET utf8 ;

USE `Accounting\_PO` ;

-- -----

-- Table `Accounting\_PO`.`Cathedra`

-- -----

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`Cathedra` (
  `idCathedra` INT NOT NULL AUTO_INCREMENT,
  `Name_cathedra` VARCHAR(15) NOT NULL,
  `Head_cathedra` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`idCathedra`),
  UNIQUE INDEX `idКафедры_UNIQUE` (`idCathedra` ASC))
ENGINE = InnoDB;

```

```

-----
-- Table `Accounting_PO`.`Auditorium`
-----

```

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`Auditorium` (
  `idAuditorium` INT NOT NULL AUTO_INCREMENT,
  `№Auditorium` INT NOT NULL,
  `Cathedra_idCathedra` INT NOT NULL,
  PRIMARY KEY (`idAuditorium`),
  INDEX `fk_Ayditoria_Cathedra1_idx` (`Cathedra_idCathedra` ASC),
  UNIQUE INDEX `idAuditorium_UNIQUE` (`idAuditorium` ASC) ,
  CONSTRAINT `fk_Ayditoria_Cathedra1`
    FOREIGN KEY (`Cathedra_idCathedra`)
    REFERENCES `Accounting_PO`.`Cathedra` (`idCathedra`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `Accounting_PO`.`Computer`
-----

```

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`Computer` (
  `idComputer` INT NOT NULL AUTO_INCREMENT,
  `№Computer` INT NOT NULL,
  `Auditorium_idAuditorium` INT NOT NULL,
  PRIMARY KEY (`idComputer`),
  UNIQUE INDEX `idComputer_UNIQUE` (`idComputer` ASC),
  INDEX `fk_Computer_Auditorium1_idx` (`Auditorium_idAuditorium` ASC),
  CONSTRAINT `fk_Computer_Auditorium1`
    FOREIGN KEY (`Auditorium_idAuditorium`)
    REFERENCES `Accounting_PO`.`Auditorium` (`idAuditorium`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `Accounting_PO`.`Licenses`
-----

```

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`Licenses` (
  `idLicenses` INT NOT NULL AUTO_INCREMENT,
  `License_type` VARCHAR(15) NOT NULL,
  PRIMARY KEY (`idLicenses`),
  UNIQUE INDEX `idЛицензии_UNIQUE` (`idLicenses` ASC))
ENGINE = InnoDB;

```

```

-----
-- Table `Accounting_PO`.`CategoriesPO`
-----

```

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`CategoriesPO` (

```

```

`idCategories_PO` INT NOT NULL AUTO_INCREMENT,
`Name_categories` VARCHAR(45) NOT NULL,
PRIMARY KEY (`idCategories_PO`),
UNIQUE INDEX `idКатегории ПО_UNIQUE` (`idCategories_PO` ASC))
ENGINE = InnoDB;

```

```

-- -----
-- Table `Accounting_PO`.`Contract`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`Contract` (
  `idContract` INT NOT NULL AUTO_INCREMENT,
  `Activation_date` DATE NOT NULL,
  `Expiration_date` DATE NOT NULL,
  PRIMARY KEY (`idContract`),
  UNIQUE INDEX `№Договора_UNIQUE` (`idContract` ASC))
ENGINE = InnoDB;

```

```

-- -----
-- Table `Accounting_PO`.`PO`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`PO` (
  `idPO` INT NOT NULL AUTO_INCREMENT,
  `Name_PO` VARCHAR(45) NOT NULL,
  `Language_PO` VARCHAR(20) NOT NULL,
  `Liwenzia_idLicenses` INT NOT NULL,
  `CategoriesPO_idCategories_PO` INT NOT NULL,
  `Contract_id_Contract` INT NOT NULL,
  PRIMARY KEY (`idPO`),

```

```

UNIQUE INDEX `idPIO_UNIQUE` (`idPO` ASC),
INDEX `fk_PO_Liwenzia1_idx` (`Liwenzia_idLicenses` ASC),
INDEX `fk_PO_CategoriesPO1_idx` (`CategoriesPO_idCategories_PO` ASC),
INDEX `fk_PO_Contract1_idx` (`Contract_id_Contract` ASC),
CONSTRAINT `fk_PO_Liwenzia1`
    FOREIGN KEY (`Liwenzia_idLicenses`)
    REFERENCES `Accounting_PO`.`Licenses` (`idLicenses`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_PO_CategoriesPO1`
    FOREIGN KEY (`CategoriesPO_idCategories_PO`)
    REFERENCES `Accounting_PO`.`CategoriesPO` (`idCategories_PO`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_PO_Contract1`
    FOREIGN KEY (`Contract_id_Contract`)
    REFERENCES `Accounting_PO`.`Contract` (`idContract`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-- -----
-- Table `Accounting_PO`.`Computer_specifications`
-- -----

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`Computer_specifications` (
    `idComputer_specifications` INT NOT NULL AUTO_INCREMENT,
    `Name` VARCHAR(45) NOT NULL,
    `Unit` VARCHAR(45) NOT NULL,

```

```

    UNIQUE INDEX `idХарактеристикиКомпьютера_UNIQUE`
(`idComputer_specifications` ASC),
    PRIMARY KEY (`idComputer_specifications`))
ENGINE = InnoDB;

```

```

-- -----
-- Table `Accounting_PO`.`RequirementsPO`
-- -----

```

```

CREATE TABLE IF NOT EXISTS `Accounting_PO`.`RequirementsPO` (
    `Value` INT NOT NULL,
    `PO_idPO` INT NOT NULL,
    `Computer_specifications_idComputer_specifications` INT NOT NULL,
    PRIMARY KEY (`PO_idPO`,
`Computer_specifications_idComputer_specifications`),
    INDEX `fk_RequirementsPO_PO1_idx` (`PO_idPO` ASC),
    INDEX `fk_RequirementsPO_Computer_specifications1_idx`
(`Computer_specifications_idComputer_specifications` ASC),
    CONSTRAINT `fk_RequirementsPO_PO1`
        FOREIGN KEY (`PO_idPO`)
        REFERENCES `Accounting_PO`.`PO` (`idPO`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
    CONSTRAINT `fk_RequirementsPO_Computer_specifications1`
        FOREIGN KEY (`Computer_specifications_idComputer_specifications`)
        REFERENCES `Accounting_PO`.`Computer_specifications`
(`idComputer_specifications`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

-- -----
-- Table `Accounting\_PO`.`Characteristic\_values`
-- -----

```
CREATE TABLE IF NOT EXISTS `Accounting_PO`.`Characteristic_values` (  
  `Value` INT NOT NULL,  
  `Computer_idComputer` INT NOT NULL,  
  `Computer_specifications_idComputer_specifications` INT NOT NULL,  
  PRIMARY KEY (`Computer_idComputer`,  
  `Computer_specifications_idComputer_specifications`),  
  INDEX `fk_Characteristic_values_Computer1_idx` (`Computer_idComputer`  
ASC),  
  INDEX `fk_Characteristic_values_Computer_specifications1_idx`  
(`Computer_specifications_idComputer_specifications` ASC),  
  CONSTRAINT `fk_Characteristic_values_Computer1`  
    FOREIGN KEY (`Computer_idComputer`)  
    REFERENCES `Accounting_PO`.`Computer` (`idComputer`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Characteristic_values_Computer_specifications1`  
    FOREIGN KEY (`Computer_specifications_idComputer_specifications`)  
    REFERENCES `Accounting_PO`.`Computer_specifications`  
(`idComputer_specifications`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- -----

-- Table `Accounting\_PO`.`PO\_computer`

-- -----

```
CREATE TABLE IF NOT EXISTS `Accounting_PO`.`PO_computer` (  
  `Computer_idComputer` INT NOT NULL,  
  `PO_idPO` INT NOT NULL,  
  PRIMARY KEY (`Computer_idComputer`, `PO_idPO`),  
  INDEX `fk_PO_computer_Computer1_idx` (`Computer_idComputer` ASC),  
  INDEX `fk_PO_computer_PO1_idx` (`PO_idPO` ASC),  
  CONSTRAINT `fk_PO_computer_Computer1`  
    FOREIGN KEY (`Computer_idComputer`)  
    REFERENCES `Accounting_PO`.`Computer` (`idComputer`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_PO_computer_PO1`  
    FOREIGN KEY (`PO_idPO`)  
    REFERENCES `Accounting_PO`.`PO` (`idPO`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

SET SQL_MODE=@OLD_SQL_MODE;

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

Приложение Б

Программные модули

1) Листинг файла index.php:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html lang="ru">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Учёт ПО</title>
</head>
<frameset rows="30%,70%">
    <frame src="top.html" name="TOP" scrolling="yes" noresize>
    <frameset >
        <frame src="content.html" name="CONTENT" scrolling="yes" noresize>
    </frameset>
</frameset>
<body>
</body>
</html>
```

2) Листинг файла top.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Навигация по сайту</title>

</head>
<style type="text/css">
    button.format {
```

```

    appearance: none;
    border: 1px groove black;
    border-radius: 5px;
    background: #052e35;
    color: #fff;
    padding: 8px 16px;
    font-size: 16px;
    height: 35px;
    width: 600px;
}

button.format1 {
    appearance: none;
    border: 1px groove black;
    border-radius: 5px;
    background: #052e35;
    color: #fff;
    padding: 8px 16px;
    font-size: 16px;
    height: 35px;
    width: 600px;
    position: absolute; top: 50px; left: 620px;
}

button.format2 {
    appearance: none;
    border: 1px groove black;
    border-radius: 5px;
    background: #052e35;
    color: #fff;
    padding: 8px 16px;
    font-size: 16px;

```

```

    height:35px;
    width:600px;
    position: absolute; top: 53px; left: 1280px;
}
button.format3 {
    appearance: none;
    border: 1px groove black;
    border-radius: 5px;
    background: #052e35;
    color: #fff;
    padding: 8px 16px;
    font-size: 16px;
    height:35px;
    width:600px;
    position: absolute; top: 87px; left: 1280px;
}
.colortext {
    color: #fff;
    text-indent: 8em;
}
.colortext1 {
    color: #fff;
    position: absolute; top: 4px; left: 1420px;
}
.button {
    position: absolute; top: 80px; left: 910px;
}
</style>
<body style="background: #052e35">

```

```

<p><a class="button" href="content.html" target="CONTENT"></a></p>
<p class="colortext"> Запросы работающие с выводом данных:</p>
<p class="colortext1">Запросы работающие с изменением данных:</p>
<form action="zapros1.php" method="post" target="CONTENT" >
    <button class="format"> ПО написанное на языке Java</button>
</form>
<form action="zapros2.php" method="post" target="CONTENT">
    <button class="format">Компьютеры подходящие под Altium по
оперативной памяти</button>
</form>
<form action="zapros3.php" method="post" target="CONTENT">
    <button class="format">Количество компьютеров с Altium по
кафедрам</button>
</form>
<form action="zapros4.php" method="post" target="CONTENT">
    <button position = "absolute" top="100" left = "620" class="format2"
>Обновление договора истекающего в 2020г</button>
</form>
<form action="zapros5.php" method="post" target="CONTENT">
    <button class="format">Договор</button>
</form>
<form action="zapros6.php" method="post" target="CONTENT">
    <button class="format3">Добавление данных</button>
</form>
<form action="zapros7.php" method="post" target="CONTENT">
    <button class="format">Нахождение min/max для занимаемой памяти по
категории ПО</button>
</form>
<form action="zapros8.php" method="post" target="CONTENT">

```

```

    <button class="format">Программные средства по аудиториям</button>
</form>
<form action="zapros9.php" method="post" target="CONTENT">
    <button class="format">ПО</button>
</form>
</body>
</html>

```

3) Листинг файла content.html:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Содержание сайта</title>
</head>
<body style="background: #052e35">
<p></p>
</body>
</html>

```

4) Листинг файла zapros1.php:

```

<html Lang="ru">
<head>
    <title>cathedra</title>
    <style type="text/css">
        TABLE {
            border: 5px double #fff; /* Рамка вокруг таблицы */
            margin: auto; /* Выравниваем таблицу по центру окна */
            color: white;
        }
        TD, TH {

```

```

padding: 5px; /* Поля вокруг текста */
border: 1px solid #fff; /* Рамка вокруг ячеек */
}
.colortext1 {
    color: #fff;
}
</style>
</head>
<body bgcolor="#052e35">
<p align="middle" class="colortext1">Результат выполнения запроса:</p>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query = "SELECT `po`.`Name_PO`, `po`.`Language_PO` FROM `po` WHERE
`po`.`Language_PO` = 'Java'";
if ($result = mysqli_query($connect,$query)){
    if ($result->num_rows) {
        echo "<table>";
        echo "<tr><th>Название ПО</th><th>Язык ПО</th></tr>";
        while ($row = $result->fetch_assoc()) {
            echo
" <tr><td>".$row['Name_PO']. "</td><td>".$row['Language_PO']. "</td></tr>";
        }
        echo "</table>";
    }
    $result->close();
}
?>
</body>
</html>

```

5) Листинг файла zapros2.php:

```

<html Lang="ru">
<head>
    <title>cathedral</title>
    <style type="text/css">
        TABLE {
            border: 5px double #fff; /* Рамка вокруг таблицы */
            margin: auto; /* Выравниваем таблицу по центру окна */
            color: white;
        }
        TD, TH {
            padding: 5px; /* Поля вокруг текста */
            border: 1px solid #fff; /* Рамка вокруг ячеек */
        }
        .colortext1 {
            color: #fff;
        }
    </style>
</head>
<body bgcolor="#052e35">
<p align="middle" class="colortext1">Результат выполнения запроса:</p>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query    = "SELECT      `computer`.`idComputer`,      `po`.`Name_PO`,
`computer_specifications`.`Name`, `characteristic_values`.`Value`
FROM `computer`
        , `po`
        INNER JOIN `requirementspo` ON `requirementspo`.`PO_idPO` =
`po`.`idPO`";

```

```

        INNER JOIN `computer_specifications` ON
`requirementspo`.`Computer_specifications_idComputer_specifications` =
`computer_specifications`.`idComputer_specifications`
        INNER JOIN `characteristic_values` ON
`characteristic_values`.`Computer_specifications_idComputer_specifications` =
`computer_specifications`.`idComputer_specifications`
WHERE `po`.`Name_PO` = 'Altium' AND `computer_specifications`.`Name` =
'ОперативнаяПамять' AND `characteristic_values`.`Value` >=
`requirementspo`.`Value`
GROUP BY idComputer";
if ($result = mysqli_query($connect,$query)){
    if ($result->num_rows) {
        echo "<table>";
        echo "<tr><th>id компьютера</th><th>Название ПО</th><th>Название
характеристики</th><th>Значение</th></tr>";
        while ($row = $result->fetch_assoc()) {
            echo
" <tr><td>".$row['idComputer']. "</td><td>".$row['Name_PO']. "</td><td>".$row['
Name']. "</td><td>".$row['Value']. "</td></tr>";
        }
        echo "</table>";
    }
    $result->close();
}
?>
</body>
</html>
6) Листинг файла zapros3.php:
<html Lang="ru">
<head>

```

```

<title>cathedral</title>
<style type="text/css">
    TABLE {
        border: 5px double #fff; /* Рамка вокруг таблицы */
        margin: auto; /* Выравниваем таблицу по центру окна */
        color: white;
    }
    TD, TH {
        padding: 5px; /* Поля вокруг текста */
        border: 1px solid #fff; /* Рамка вокруг ячеек */
    }
    .colortext1 {
        color: #fff;
    }
</style>
</head>
<body bgcolor="#052e35">
<p align="middle" class="colortext1">Результат выполнения запроса:</p>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query = "SELECT `cathedral`.`Name_cathedral`, `po`.`Name_PO`, COUNT(*)
FROM `cathedral`
        INNER JOIN `auditorium` ON `auditorium`.`Cathedral_idCathedral` =
`cathedral`.`idCathedral`
        INNER JOIN `computer` ON `computer`.`Auditorium_idAuditorium` =
`auditorium`.`idAuditorium`
        INNER JOIN `po_computer` ON `po_computer`.`Computer_idComputer` =
`computer`.`idComputer`
        INNER JOIN `po` ON `po_computer`.`PO_idPO` = `po`.`idPO`
WHERE `po`.`Name_PO` = 'Altium'

```

```

GROUP BY Name_cathedral";
if ($result = mysqli_query($connect,$query)){
    if ($result->num_rows) {
        echo "<table>";
        echo          "<tr><th>Название          кафедры</th><th>Название
ПО</th><th>Количество</th></tr>";
        while ($row = $result->fetch_assoc()) {
            echo
            "<tr><td>".$row['Name_cathedral']."</td><td>".$row['Name_PO']."</td><td>".$r
ow['COUNT(*)']."</td></tr>";
        }
        echo "</table>";
    }
    $result->close();
}
?>
</body>
</html>

```

7) Листинг файла zapros4.php:

```

<html Lang="ru">
<head>
    <title>cathedral</title>
    <style type="text/css">
        TABLE {
            border: 5px double #fff; /* Рамка вокруг таблицы */
            margin: auto; /* Выравниваем таблицу по центру окна */
            color: white;
        }
        TD, TH {
            padding: 5px; /* Поля вокруг текста */

```

```

        border: 1px solid #fff; /* Рамка вокруг ячеек */
    }
    .colortext1 {
        color: #fff;
    }
</style>
</head>
<body bgcolor="#052e35">
<p align="middle" class="colortext1">Договор был успешно обновлён</p>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query    = "UPDATE `contract` SET `contract`.`Expiration_date` =
DATE_ADD(`Expiration_date`, INTERVAL 1 YEAR)
WHERE (((Year(`contract`.`Expiration_date`))=2020))";
if ($result = mysqli_query($connect,$query)){

}
?>
</body>
</html>

```

8) Листинг файла zapros5.php:

```

<html Lang="ru">
<head>
    <title>cathedra</title>
    <style type="text/css">
        TABLE {
            border: 5px double #fff; /* Рамка вокруг таблицы */
            margin: auto; /* Выравниваем таблицу по центру окна */
            color: white;
        }
    </style>

```

```

    TD, TH {
        padding: 5px; /* Поля вокруг текста */
        border: 1px solid #fff; /* Рамка вокруг ячеек */
    }
.colortext1 {
    color: #fff;
}
</style>
</head>
<body bgcolor="#052e35">
<p align="middle" class="colortext1">Результат выполнения запроса:</p>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query = "SELECT * FROM `contract`";
if ($result = mysqli_query($connect,$query)){
    if ($result->num_rows) {
        echo "<table>";
        echo "    <tr><th>id    договора</th><th>Дата    активации</th><th>Дата
окончания</th></tr>";
        while ($row = $result->fetch_assoc()) {
            echo
"    <tr><td>".$row['idContract']. "</td><td>".$row['Activation_date']. "</td><td>".$r
ow['Expiration_date']. "</td></tr>";
        }
        echo "</table>";
    }
    $result->close();
}
?>
</body>

```

```
</html>
```

9) Листинг файла zapros6.php:

```
<html Lang="ru">
<head>
  <title>cathedra</title>
  <style type="text/css">
    TABLE {
      border: 5px double #fff; /* Рамка вокруг таблицы */
      margin: auto; /* Выравниваем таблицу по центру окна */
      color: white;
    }
    TD, TH {
      padding: 5px; /* Поля вокруг текста */
      border: 1px solid #fff; /* Рамка вокруг ячеек */
    }

  </style>
</head>
<body bgcolor="#052e35">
<p align="middle"> <font color="white">Ведите данные для добавления
</font></p>
<form method="post">
  <font color="white"> Название</font>
  <input type="text" name="Name_PO"/>
  <font color="white"> Язык ПО </font>
  <input type="text" name="Language_PO"/>
  <font color="white">id Лицензии</font>
  <input type="text" name="Liwenzia_idLicenses"/>
  <font color="white">id Категории ПО</font>
```

```

<input type="text" name="CategoriesPO_idCategories_PO"/>
<font color="white">id Договора</font>
<input type="text" name="Contract_id_Contract"/>
<input type="submit" name="send" value="Добавить"/>
</form>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query = "SELECT `po`.* FROM po";
if (isset($_POST['send'])) {
    $Name_PO = $_POST['Name_PO'];
    $Language_PO = $_POST['Language_PO'];
    $Liwenzia_idLicenses = $_POST['Liwenzia_idLicenses'];
    $CategoriesPO_idCategories_PO =
$_POST['CategoriesPO_idCategories_PO'];
    $Contract_id_Contract = $_POST['Contract_id_Contract'];
    mysqli_query($connect, "INSERT INTO `po`
(Name_PO`,`Language_PO`,`Liwenzia_idLicenses`,`CategoriesPO_idCategories_
PO`,`Contract_id_Contract`) VALUES
('$Name_PO','$Language_PO','$Liwenzia_idLicenses','$CategoriesPO_idCategori
es_PO','$Contract_id_Contract')");
}
?>
</body>
</html>

```

10) Листинг файла zapros7.php:

```

<html Lang="ru">
<head>
<title>cathedra</title>
<style type="text/css">
TABLE {

```

```

border: 5px double #fff; /* Рамка вокруг таблицы */
margin: auto; /* Выравниваем таблицу по центру окна */
color: white;
}
TD, TH {
padding: 5px; /* Поля вокруг текста */
border: 1px solid #fff; /* Рамка вокруг ячеек */
}
</style>
</head>
<body bgcolor="#052e35">
<p align="middle"><font color="white">Результат выполнения
запроса:</font></p>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query = "SELECT
`categoriespo`.`Name_categories`,MAX(VALUE),MIN(VALUE)
FROM `categoriespo`
INNER JOIN `po` ON
`po`.`CategoriesPO_idCategories_PO`=`categoriespo`.`idCategories_PO`
INNER JOIN `requirementspo` ON `requirementspo`.`PO_idPO` = `po`.`idPO`
WHERE `requirementspo`.`Computer_specifications_idComputer_specifications`
= 3
GROUP BY Name_categories";
if ($result = mysqli_query($connect,$query)){
if ($result->num_rows) {
echo "<table>";
echo "<tr><th>Название
категории</th><th>Max,GB</th><th>Min,GB</th></tr>";
while ($row = $result->fetch_assoc()) {

```

```

        echo
        "<tr><td>".$row['Name_categories']."</td><td>".$row['MAX(VALUE)']."</td><t
        d>".$row['MIN(VALUE)']."</td></tr>";
    }
    echo "</table>";
}
$result->close();
}
?>
</body>
</html>

```

11) Листинг файла zapros8.php:

```

<html Lang="ru">
<head>
    <title>cathedral</title>
    <style type="text/css">
        TABLE {
            border: 5px double #fff; /* Рамка вокруг таблицы */
            margin: auto; /* Выравниваем таблицу по центру окна */
            color: white;
        }
        TD, TH {
            padding: 5px; /* Поля вокруг текста */
            border: 1px solid #fff; /* Рамка вокруг ячеек */
        }
    </style>
</head>
<body bgcolor="#052e35">
<p align="middle"><font color="white">Результат выполнения
запроса:</font></p>

```

```

<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query = "SELECT `auditorium`.`№Auditorium`, `po`.`Name_PO` FROM
`auditorium`, `po` ORDER BY `auditorium`.`№Auditorium` ASC";
if ($result = mysqli_query($connect,$query)){
    if ($result->num_rows) {
        echo "<table>";
        echo "<tr><th>№ аудитории</th><th>Название ПО</th></tr>";
        while ($row = $result->fetch_assoc()) {
            echo
            "<tr><td>".$row['№Auditorium']. "</td><td>".$row['Name_PO']. "</td></tr>";
        }
        echo "</table>";
    }
    $result->close();
}
?>
</body>
</html>

```

12) Листинг файла zapros9.php:

```

<html Lang="ru">
<head>
    <title>cathedra</title>
    <style type="text/css">
        TABLE {
            border: 5px double #fff; /* Рамка вокруг таблицы */
            margin: auto; /* Выравниваем таблицу по центру окна */
            color: white;
        }
        TD, TH {

```

```

padding: 5px; /* Поля вокруг текста */
border: 1px solid #fff; /* Рамка вокруг ячеек */
}
</style>
</head>
<body bgcolor="#052e35">
<p align="middle"><font color="white">Результат выполнения
запроса:</font></p>
<?php
$connect = mysqli_connect('localhost', 'root', '', 'accounting_po');
$query = "SELECT * FROM `po`";
if ($result = mysqli_query($connect,$query)){
    if ($result->num_rows) {
        echo "<table>";
        echo "<tr><th>id ПО</th><th>Имя ПО</th><th>Язык ПО</th><th>id
Лицензии</th><th>id Категории ПО</th><th>id Договора</th></tr>";
        while ($row = $result->fetch_assoc()) {
            echo
"<tr><td>".$row['idPO']. "</td><td>".$row['Name_PO']. "</td><td>".$row['Langu
age_PO']. "</td><td>".$row['Liwenzia_idLicenses']. "</td><td>".$row['Categories
PO_idCategories_PO']. "</td><td>".$row['Contract_id_Contract']. "</td></tr>";
        }
        echo "</table>";
    }
    $result->close();
}
?>
</body>
</html>

```